COLLABORATIVE PROTECTION NETWORK FOR THE DETECTION OF FLOODING DDOS ATTACKS

[1]THULLIBILLI.SREEKANTH, [2] JELDI.SAGAR BABU
[1]Assistant Professor, [2]Assistant professor
[1, 2] Computer Science and Engineering
[1]Narsimha Reddy Engineering College, Maisammaguda, Hyderabad
[2]Princeton College of Engineering and Technology, Ghatkesar, Hyderabad

A**bstract** — Distributed denial-of-service (DDoS) attacks remain a major security problem, the mitigation of which is very hard especially when it comes to highly distributed botnet-based attacks. The early discovery of these attacks, although challenging, is necessary to protect end-users as well as the expensive network infrastructure resources. In this paper, we address the problem of DDoS attacks and present the theoretical foundation, architecture, and algorithms of FireCol. The core of FireCol is composed of intrusion prevention systems (IPSs) located at the Internet service providers (ISPs) level. The IPSs form virtual protection rings around the hosts to defend and collaborate by exchanging selected traffic information. The evaluation of FireCol using extensive simulations and a real dataset is presented, showing FireCol effectiveness and low overhead, as well as its support for incremental deployment in real networks.

I. INTRODUCTION

Distributed denial-of-service (DDoS) attacks still constitute a major concern even though any works have tried to address this issue in the past (ref. survey in [2]). As they evolved from relatively humble megabit beginnings in 2000, the largest DDoS attacks have now grown a hundredfold to break the 100 gb/s, for which the majority of ISPs today lack an appropriate infrastructure to mitigate them.

Most recent works aim at countering DDoS attacks by fighting the underlying vector, which is usually the use of botnets .

A botnet is a large network of compromised machines (bots) controlled by one entity (the master). The master can launch synchronized attacks, such as DDoS, by sending orders to the bots via a Command & Control channel. Unfortunately, detecting a botnet is also hard, and efficient solutions may require to participate actively to the botnet itself [4], which raises important ethical issues, or to first detect botnet-related malicious activities (attacks, infections, etc.), which may delay the mitigation.

To avoid these issues, this paper focuses on the detection of DDoS attacks and per se not their underlying vectors. Although non distributed denial of service attacks usually exploit vulnerability by sending few carefully forged packets to disrupt a service, DDoS attacks are mainly used for flooding a particular victim with massive traffic c as highlighted.In fact, the popularity of these attacks is due to their high effectiveness against any kind of service since there is no need to identify and exploit any particular service-specific flaw in the victim. Hence, this paper focuses exclusively on flooding DDoS attacks.

A single intrusion prevention system (IPS) or intrusion detection system (IDS) can hardly detect such DDoS attacks, unless they are located very close to the victim. However, even in that latter case, the IDS/IPS may crash because it needs to deal with an overwhelming volume of packets (some flooding attacks reach 10–100 Gb/s) . In addition, allowing such huge traffic to transit through the Internet and only detect/block it at the host IDS/IPS may severely strain Internet resources.

This paper presents FireCol, a new collaborative system that detects flooding DDoS attacks as far as possible from the victim host and as close as possible to the attack source(s) at the Internet service provider (ISP) level. FireCol relies on a distributed architecture composed of multiple IPSs forming overlay networks of protection rings around subscribed customers.

FireCol is designed in a way that makes it a service to which customers can subscribe. Participating IPSs along the path to a subscribed customer collaborate (vertical communication) by computing and exchanging belief scores on potential attacks. The IPSs form virtual protection rings around the host they protect. The virtual rings use horizontal communication when the degree of a potential attack is high. In this way, the threat is measured based on the overall traffic bandwidth directed to the customer compared to the maximum bandwidth it supports.

II. RELATED WORK

Paper describes a preliminary architecture of FireCol with initial simulations. In this paper, these are substantially extended by enhancing and detailing the communication algorithms. A mitigation technique is provided as well as a detailed investigation of FireCol configuration. Experimentation with a real dataset and different traffic patterns was also performed. Even though a publicly available dataset was used, this does not ease the quantitative comparison to related work. Unlike packet-based methods, false and true positives are computed globally taking into account each router and each time window.

Bellovin proposes in the use of distributed firewalls. However, only firewall rules are exchanged, and each firewall must detect the attacks on its own. The authors propose a similar solution where a Gateway is requested to block the traffic of an attack. only the DDoS mitigation of the attacks is distributed, but the detection is located very close to the victim. Unlike FireCol, all previously mentioned solutions do not exploit effective use of collaboration.

The approach is based on content-filtering. a peer-to-peer approach is introduced, mobile-agents are leveraged to exchange newly detected threats. FireCol provides a simpler solution in the sense that it uses simple metrics, while the former approaches can be costly in terms of resource consumption. Other approaches promoting the use of simple statistics are not distributed. Mahajan et al. introduce a technique for detecting overloaded links based on traffic aggregation. Belief functions are also used by Peng et al. in to detect DDoS attacks based on counting new IP addresses. These works are close but differ from FireCol, in which detection is focused on the potential victim. The authors in dealt with DoS related overload issues by a cluster architecture to analyze firewall observations.

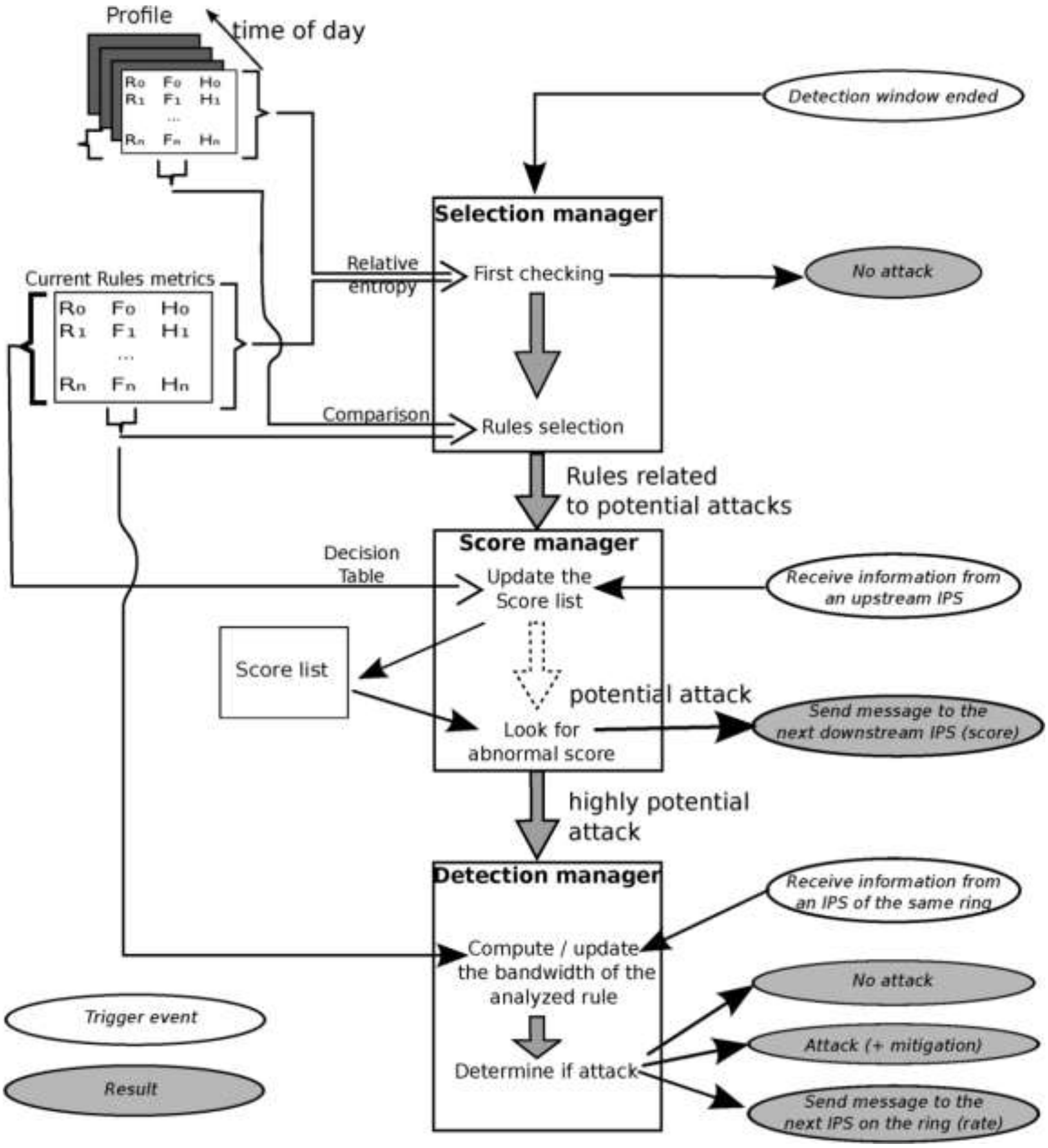


Fig. 1. FireCol architecture.

In a DoS resistant communication mechanism is proposed for end-hosts by using acknowledgments. Another solution relies on tokens delivered to each new TCP flow. Each router between the source and the destination marks the path to detect spoofed addresses. Detection of specific SYN flooding attacks at the router level is investigated .

The correlation between the requests and replies to detect flooding attacks to limit overhead. The observation of past attacks or legitimate traffic in order to create a community of interest is another alternative . Information sharing about DDoS attacks is also addressed, but from a high-level perspective where a trusted network of partners (networks) is built. Detecting DDoS attacks by detecting IP spoofing is addressed and is related to our work as the goal is to speed up and limit the costs of packet filtering, especially in the case of DoS attack. Moreover, statistics on the network traffic are used like the entropy . There are also DDoS countering techniques dedicated to specific applications such as Web servers or clouds. Detecting the DDoS attacks at the ISP level was also studied, but these approaches analyze all traffic, unlike FireCol, which is based on a local mechanism enhanced by the collaboration when needed. Shares information between different network nodes to mitigate efficiently flooding attacks, FireCol leverages ring semantic in order to enhance the analysis of shared information.

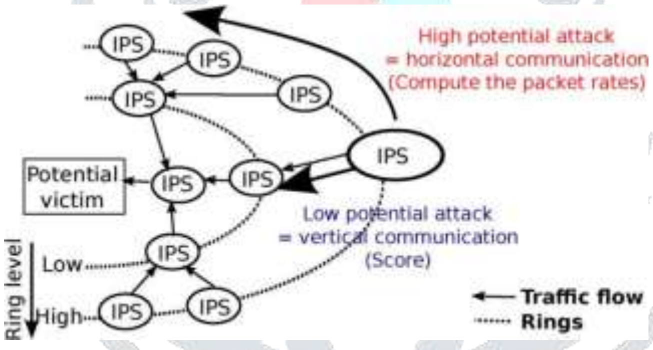


Fig. 2. Horizontal and vertical communication in FireCol.

III. FIRECOL ARCHITECTURE

A. ring-based overlay protection

The FireCol system (Fig. 1) maintains virtual rings or shields of protection around registered customers. A ring is composed of a set of IPSs that are at the same distance (number of hops) from the customer (Fig. 2). As depicted in Fig. 1, each FireCol IPS instance analyzes aggregated traffic within a configure able detection window. The metrics manager computes the frequencies and the entropies of each rule (Section III-A). A rule de-scribes a specific traffic instance to monitor and is essentially a traffic filter, which can be based on IP addresses or ports.

Following each detection window, the selection manager measures the deviation of the current traffic profile from the stored ones, selects out of profile rules, and then forwards them to the score manager. Using a decision table, the score manager assigns a score to each selected rule based on the frequencies, the entropies, and the scores received from Upstream IPSs (vertical collaboration/communication). Using a threshold, a quite low score is marked as a low potential attack and is communicated to the downstream IPS that will use to compute its own score. A quite high score on the other hand is marked as high potential attack and triggers ring-level (horizontal) communication (Fig. 2) in order to confirm or dismiss the attack based on the computation of the actual packet rate crossing the ring surpasses the known, or evaluated, customer capacity (Section II -B).

As can be noticed, this detection mechanism inherently generates no false positives since each potential attack is checked. However, since the entire traffic cannot be possibly monitored, we promote the usage of multiple levels and collaborative filtering described previously for an efficient selection of rules, and so traffic, along the process. In brief, to save resources, the collaboration manager is only invoked for the few selected candidate rules based on resource-friendly metrics.

B. subscription protocol

FireCol protects subscribers (i.e., potential victims) based on defined rules. A FireCol rule matches a pattern of IP packets. Generally, this corresponds to an IP sub network or a single IP address. However, the rule definition can include any other monitor able information that can be monitored, such as the protocols or the ports used.

FireCol is an added value service to which customers subscribe using the protocol depicted in Fig. 3. The protocol uses a trusted server of the ISP that issues tokens. When a customer subscribes for the FireCol protection service, the trusted server using a two phase process. First, the router sends a message RMsg to the protected customer containing a counter initialized to 0. The counter is incremented each time it passes through a FireCol-enabled router. The customer (or first -level FireCol router) then replies to the initiating router with the value of its ring level. This procedure is optimized through aggregation when several routers are requesting a ring-level update.

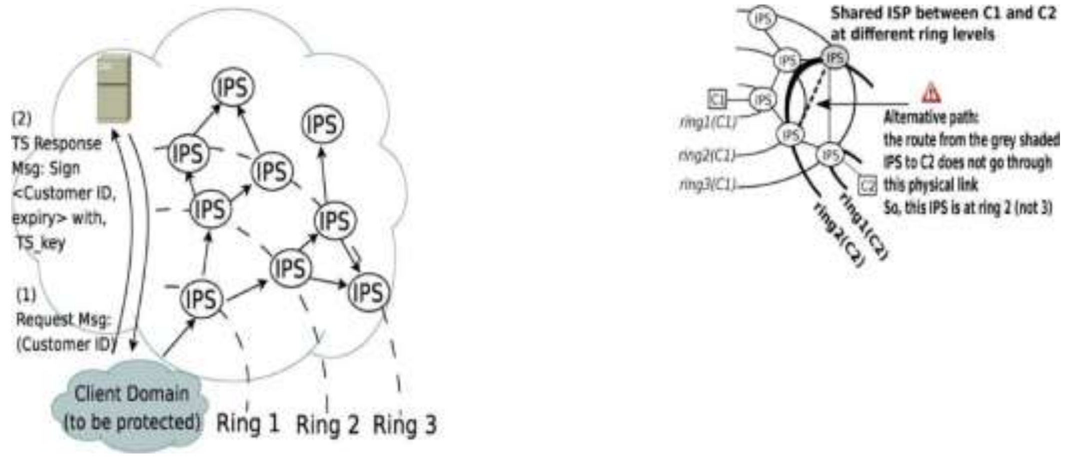


Fig. 3. FireCol subscription protocol

adds an entry with the subscribing rule along with its subscription period (TTL) and the supported capacity. The server then issues periodically a corresponding token to the customer with a TTL and a unique ID signed using its private key. All communications between subscribers and the server are secured a using private/public key encryption

In practice, the ring level value is network-dependent. However, routing stability has been well investigated and enhanced. The study done It shows that most routes are usually stable within the order of several days, while flooding attacks generally operate within the order of minutes in order to have a high impact. For further analysis, Section VI-I quantifies the impact of routers not assigned to the right level. It shows that updating the ring topology at regular intervals is sufficient even if some IPSs are not well configured with respect to the ring to which they belong. A more sophisticated mechanism could monitor route changes to force ring updates.

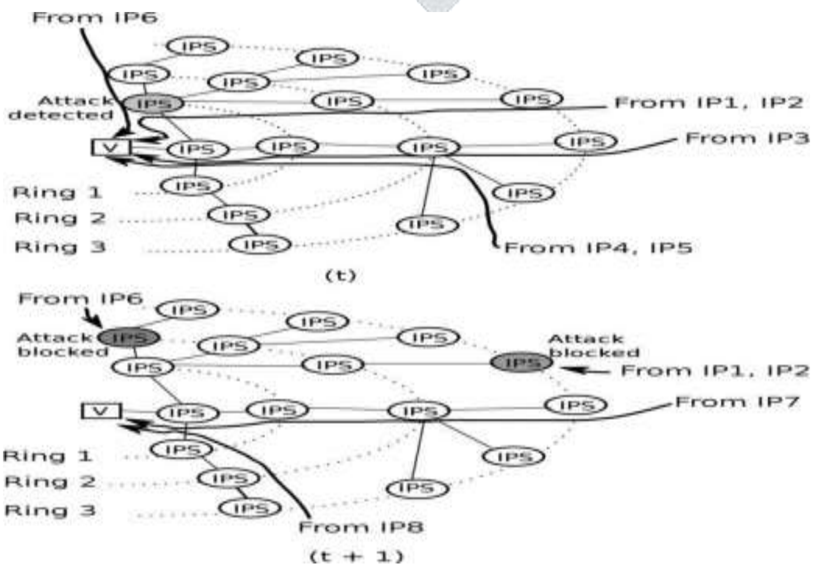
In FireCol, a capacity is associated to each rule. Rule capacities can be provided either by customers or the ISP (for overall capacity rules). For sensitive services, customers can specify the capacity. IT services of large companies should be able to provide such information regarding their infrastructure. For smaller customers, statistical or learning algorithms, running at customer premises or first hop IPS, might be leveraged to pro-file traffic throughput [9]. Similar to [10], the threshold can be tuned to keep a small proportion (i.e., 5%) for analysis. Finally, for very small customers, such as a household, a single rule related to the capacity of the connection can be used. The maximum capacity, or throughput quota, is generally readily available to the ISP based on the customer service level agreement (SLA).

C. multiple customers

Because of their inherent complete independence, FireCol allows the coexistence of multiple virtual protection rings for multiple customers across the same set of IPSs. Therefore, a single IPS may act at different levels with respect the customers it protects as depicted in Fig. 4. Although most of the figures in this paper represent overlay networks with a single route, from an ISP to a customer, this figure highlights that alternative paths are possible. However, as discussed in the previous section, the rings are dependent of the routing at a certain time, which is quite stable compared to the typical duration of flooding attacks, and so only the current route is considered for building the rings.

For each selected r_i , the collaboration manager computes the corresponding packet rate using rule frequencies and the overall bandwidth (bw_w) consumed during the last detection window. If the rate is higher than the rule capacity cap_i , an alert is raised. Otherwise, the computed rate is sent to the next IPS on the ring (Algorithm 1). When an IPS receives a request to calculate the aggregate packet rate for a given rule, it first checks if it was the initiator. In this case, it deduces that the request has already made the round of the ring, and hence there is no potential attack. Otherwise, it calculates the new rate by adding in its own rate and checking if the maximum capacity is reached, in which case an alert is raised. Otherwise, the investigation is delegated to the next horizontal IPS on the ring.

The first IPS fills it and sets the Boolean I_r to true (line 16). I_r is reset after the computation finishes, i.e., when the request has made the round of the ring or when the alert is triggered. With simple adjustments, ring traversal overhead can further be reduced if several suspect rules are investigated in one pass. Rate computation can be performed based on the number of packets per second (pps) or bytes per second (bps). The first method is more suitable for detecting flooding DDoS attacks having a small packet pattern, such as SYN floods. Bytes -based method is better for detecting flooding attacks with large packet payloads. FireCol customers can subscribe to either or both protection types.



(t)

(t + 1)

Fig. 4. At the end of time t, an attack against host V is detected. At time $i+1$, the traffic from attack sources is blocked.

IV. FIRECOL ATTACK DETECTION ALGORITHMS

When an attack is detected, FireCol rings form protection shields around the victim. In order to block the attack as close as possible to its source(s), the IPS that detects the attack informs its upper-ring IPSs (upstream IPSs), which in turn apply the vertical communication process and enforce the protection at their ring level (Algorithm 2). To extend the mitigation, the IPS that detects the attack informs also its peer IPSs on the same ring to block traffic For each selected r_i , the collaboration manager computes the corresponding packet rate using rule frequencies and the overall bandwidth (b_{win}) consumed during the last detection window. If the rate is higher than the rule capacity cap_i , an alert is raised. Otherwise, the computed rate is sent to the next IPS on the ring (Algorithm 1).

Algorithm 1: checkRule (IPS_id, i, $rate_i$, cap_i)

```
1: if $b_i \wedge (IPS\_id \neq null)$ then
2:     if $IPS\_id == myID$ then
3:         $b_i \leftarrow false$;
4:         return
5:     else
6:         $rate_i \leftarrow rate_i + F_i$
7:         if $rate_i > cap_i$ then
8:             $b_i \leftarrow false$;
9:             raise DDOS alert;
10:            return
11:        else
12:            $nextIPS.checkRule(IPS\_$
13:        end if
14:    end if
15: else
16:    $b_i \leftarrow true$;
17:    $nextIPS.checkRule(myID, 0)$
18: end if
```

related to the corresponding rule. This is done by forwarding the information in the same manner as done by the collaboration manager (Algorithm 1). Only traffic from suspected sources (i.e., triggered some rule r_i) is blocked as shown in Fig. 4

This process entails the potential blocking of benign addresses. However, this is a temporary cost that is difficult to avoid if a flooding attack is to be stopped. Potential alternatives are described in the next section. It may be impossible to determine all attack sources during a single detection window due to inherent network delays and/or resource limitations. The attacker can also invoke an attack scenario from different machines at different times to reduce the risk of detection.

For this, after the detection and mitigation of an attack against some host h, FireCol continues the detection process looking for some additional attack sources. Furthermore, in order to limit the effect of potentially additional attack sources, after the blocking period elapses, the IPS may activate a cautious mode phase wherein a rate limitation of packets corresponding to the triggered rule is applied. The actual duration of the blocking and caution period depends on the aggressiveness of the attack, i.e., on the difference between the observed packet rate $rate_i$ and the host capacity.

A. careful mitigation

This section gives an overview of common techniques to improve attack mitigation by blocking only attacks- related IP sources. Only those associated to high packet rates or that have opened most of the sessions recently might be blocked like in. Moreover, identifying not-yet-seen IP addresses is another way to detect the potential spoofed addresses or zombies used to perform a DDoS attack. The authors in propose other heuristics based on the difference between incoming and outgoing traffic. A solution could be to capture all traffic associated with a triggered alert by the score manager and use signatures to clearly identify an attack. A general blacklist can be imported from external databases, like SpamHaus, which stores IP addresses related to Spam, meaning that they are probably zombie computers.

V.MITIGATION

A. mitigation shields

When an attack is detected, FireCol rings form protection shields around the victim. In order to block the attack as close as possible to its source(s), the IPS that detects the attack informs its upper-ring IPSs (upstream IPSs), which in turn apply the vertical communication process and enforce the protection at their ring level (Algorithm 2). To extend the mitigation, the IPS that detects the attack informs also its peer IPSs on the same ring to block traffic related to the corresponding rule. This is done by forwarding the information in the same manner as done by the collaboration manager (Algorithm 1). Only traffic from suspected sources (i.e., triggered some rule r_i) is blocked as shown in Fig. 7. This is performed by the block_IPs function in Algorithm 2, line 5.

Algorithm 2: mitigate (r_i , firstRing)

```
1: for all $IPS \in upstreamIPSs$ do
2:     $IPS.mitigate(r_i, False)$
```

```
end for
for all $n \in getAddr(r)$ do
    block_IPs($n$)
end for
if $firstRing = True$ then
    $nextIPS.notify(r, True)$
end if
setCautiousMode($r$)
```

VI . EVALUATION

The objective of the experiments is to evaluate the accuracy of FireCol in different configurations. Furthermore, the robust-ness of FireCol is evaluated in abnormal situations such as the existence of no cooperative routers or configuration errors.

Although obtaining real router traces is possible, getting synchronized traffic and host states of a real network along with its detailed topology is quite difficult for security, privacy, and legal reasons. Thus, we mainly used a simulation-based approach for the evaluation of the FireCol system.

VII . CONCLUSION AND FUTURE WORKS

This paper proposed FireCol, a scalable solution for the early detection of flooding DDoS attacks. Belief scores are shared within a ring-based overlay network of IPSs. It is performed as close to attack sources as possible, providing a protection to subscribed customers and saving valuable network resources. Experiments showed good performance and robustness of FireCol and highlighted good practices for its configuration. Also, the analysis of FireCol demonstrated its light computational as well as communication overhead. Being offered as an added value service to customers, the accounting for FireCol is therefore facilitated, which represents a good incentive for its deployment by ISPs. As a future work, it is planned to extend FireCol to support different IPS rule structures.

VIII. REFERENCES

[1]A. Networks, Arbor, Lexington, MA,Worldwide ISP security report," Tech. Rep., 2010.

[2]T. Peng, C. Leckie, and K. Ramamohanarao, Survey of net-work-based defense mechanisms countering the DoS and DDoS problems,"compute surv. Vol.39,apr 2007 Article 3.

[3] E. Cooke, F. Jahanian, and D. Mcpherson, "The zombie roundup: Un-derstanding, detecting, and disrupting botnets," in Proc. SRUTI, Jun. 2005,

[4]T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measure-ments and mitigation of peer-to-peer-based botnets: A case study on storm worm," in Proc. USENIX LEET, 2008, Article no. 9.

[5]J. Françcois, A. El Atawy, E. Al Shaer, and R. Boutaba, “A collabo-rative approach for proactive detection of distributed denial of service attacks,” in Proc. IEEE MonAM, Toulouse, France, 2007, vol. 11.

[6]A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, “Lo-cating Internet routing instabilities,” Comput. Commun. Rev., vol. 34, no. 4, pp. 205–218, 2004.

[7]A. Basu and J. Riecke, “Stability issues in OSPF routing,” in Proc. ACM SIGCOMM , 2001, pp. 225–236.

[8]V. Paxson, “End-to-end routing behavior in the Internet,” IEEE/ACM Trans. Netw., vol. 5, no. 5, pp.601–615, Oct. 1997.

[9]K. Xu, Z.-L. Zhang, and S. Bhattacharyya, “Internet traffic behavior profiling for network security monitoring,” IEEE/ACM Trans. Netw., vol. 16, no. 6, pp. 1241–1252, Dec. 2008.

[10] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian, “Optimizing cost and performance in online service provider networks,” in Proc. USENIX NSDI, 2010, p. 3.

[10]M. Dischinger, A. Mislove, A. Haeberlen, and K.P. Gummadi, “De-tecting bittorrent blocking,” in Proc. ACM SIGCOMM Conf. Internet Meas., 2008, pp. 3–8. Y. Zhang, Z. M. Mao, and M. Zhang, “Detecting traffic differentiation in backbone ISPs with NetPolice,” in Proc. ACM SIGCOMM Conf.